

Digital Currency, Capacity Building and Development Program

Duration: 3 Weeks

Week 1 – Developing Central Bank Digital Currency (CBDC) Solutions

Day 1: CBDC Fundamentals & Use Cases

Module 1: Introduction to CBDC

- CBDC Foundations
- Retail vs wholesale CBDC models
- Global CBDC landscape & case studies
- Reserve Bank of India e-Rupee model
- Regulatory & policy considerations
- CBDC vs cash vs commercial bank money

Lab Exercises:

1. Map existing payment systems vs proposed CBDC model for your economy
2. Analyse RBI e-Rupee pilot: design choices and lessons learned

Module 2: CBDC Use Cases & Design Considerations

- Identifying domestic & cross-border use cases
- Selecting a use case for prototype development
- CBDC design dimensions: programmability, privacy, interoperability
- Two-tier vs direct CBDC architecture
- Token-based vs account-based CBDC
- CBDC design requirements documentation

Lab Exercises:

3. Select and document a CBDC use case for prototype using a structured template
4. Draft high-level design requirements for chosen use case

Day 2: Technology Selection & System Architecture

Module 3: Technology Stack for CBDC

- Blockchain vs DLT vs centralised ledger
- Permissioned DLT platforms: Hyperledger, Corda, Quorum
- Smart contract platforms for CBDC
- Front-end tech: mobile wallet, merchant app
- Back-end: core ledger, settlement engine, APIs
- Selecting technology for back end and front end

Lab Exercises:

5. Compare Hyperledger Fabric vs Corda for CBDC issuance use case
6. Design technology stack diagram for selected CBDC prototype

Module 4: CBDC System Architecture & Issuance Design

- Designing CBDC currency issuance system
- Issuance, distribution, and redemption lifecycle
- Wallet infrastructure and key management

- Interoperability with existing payment rails
- Security architecture & threat modelling
- API design for CBDC ecosystem

Lab Exercises:

7. Draw end-to-end CBDC issuance architecture with component roles
8. Define API contract for CBDC wallet-to-merchant transaction

Day 3: CBDC Development & Simulation

Module 5: Developing CBDC – Prototype Build

- Setting up CBDC development environment
- Implementing token issuance smart contract
- Building wallet registration & onboarding
- P2P transfer and merchant payment flows
- Offline payment capability design
- Unit testing CBDC smart contracts

Lab Exercises:

9. Deploy a CBDC token smart contract on a permissioned testnet
10. Build and test a basic wallet-to-wallet transfer transaction

Module 6: CBDC Simulation & Evaluation

- Simulating CBDC in a sandbox environment
- Load testing & performance benchmarking
- AML/CFT controls simulation
- Cross-border CBDC simulation (mBridge model)
- Evaluating results against design requirements
- Lessons learned & next steps

Lab Exercises:

11. Run end-to-end CBDC simulation: issuance → transfer → redemption
12. Present simulation results and identify gaps vs requirements

Week 2 – Developing Cryptocurrency & Monitoring Solutions

Day 4: Cryptocurrency Fundamentals & Use Cases

Module 7: How Cryptocurrency Works

- Blockchain fundamentals: blocks, chains, consensus
- How Bitcoin is developed & mined
- Ethereum & smart contract layer
- Public vs permissioned blockchain
- Cryptocurrency use cases in the Africa context
- Selecting a use case for crypto development

Lab Exercises:

13. Trace a Bitcoin transaction from initiation to confirmation on a block explorer
14. Identify 3 viable crypto use cases for African financial inclusion

Module 8: Crypto Product Requirements & Technology Selection

- Documenting cryptocurrency product requirements
- Selecting front-end & back-end technology
- Wallet types: custodial vs non-custodial
- Key management & seed phrase security

- Exchange integration & liquidity considerations
- Regulatory landscape for crypto in Africa

Lab Exercises:

15. Write a product requirements document for a selected crypto use case
16. Compare custodial vs non-custodial wallet architectures for your use case

Day 5: Designing & Developing Cryptocurrency Solutions

Module 9: Designing Cryptocurrency Issuance System

- Token standards: ERC-20, ERC-721, BEP-20
- Designing coin/token issuance mechanism
- Smart contract design for token lifecycle
- Governance & upgrade patterns
- Gas optimisation strategies
- Security: reentrancy, overflow, access control

Lab Exercises:

17. Design an ERC-20 token for a selected African payment use case
18. Audit a sample smart contract for common vulnerabilities

Module 10: Developing & Simulating Cryptocurrency

- Setting up Hardhat/Foundry development environment
- Writing and deploying token smart contracts
- Building a simple web3 front end
- Simulating cryptocurrency transactions on testnet
- Integration testing across wallet and contract
- Preparing for mainnet deployment

Lab Exercises:

19. Deploy ERC-20 token on Ethereum testnet and perform test transactions
20. Run simulation: token issuance → transfer → burn on local fork

Day 6: Cryptocurrency Monitoring Tools

Module 11: Monitoring Tool Fundamentals

- Why crypto monitoring matters
- Types of monitoring tools: on-chain analytics, AML, market
- Examples of monitoring tools: Chainalysis, Elliptic, TRM Labs
- Steps for developing a custom monitoring tool
- Data sources: node RPC, block explorers, exchange feeds
- Selecting technology for monitoring tool back end and front end

Lab Exercises:

21. Explore Chainalysis KYT demo environment and identify flagged transactions
22. Design data architecture for a custom crypto monitoring dashboard

Module 12: Building & Simulating Monitoring Tools

- Documenting monitoring tool requirements
- Designing the monitoring tool system
- Building transaction ingestion pipeline
- Rule-based alert engine: threshold, velocity, pattern
- Dashboard for real-time transaction visualisation
- Simulating cryptocurrency monitoring tool

Lab Exercises:

23. Build a transaction monitoring pipeline ingesting testnet data via RPC
24. Simulate flagging of suspicious transaction patterns using rule engine

Week 3 – Developing Stable-Coin Based Solutions

Day 7: Stablecoin Fundamentals & Platform Overview

Module 13: Overview of Stablecoin Platforms

- What is a stablecoin & types: fiat, crypto, algorithmic
- Examples: USDC, USDT, DAI, eNaira, e-CFA
- Stablecoin vs CBDC: key differences
- Use cases in Africa: remittances, trade, DeFi
- Selecting a use case for stablecoin development
- Regulatory considerations for stablecoins

Lab Exercises:

25. Compare USDC (fiat-backed) vs DAI (crypto-backed) design architectures
26. Identify a stablecoin use case suitable for an African cross-border payment

Module 14: Steps & Examples of Stablecoin Platform Development

- Overview of stablecoin platform architecture
- Steps for developing stablecoin platform
- Examples of stablecoin platforms & lessons learned
- Collateral management & reserve proof systems
- Peg stability mechanisms
- Documenting stablecoin requirements

Lab Exercises:

27. Document functional and non-functional requirements for selected stablecoin
28. Review USDC reserve attestation report and map to design requirements

Day 8: Technology Selection & Stablecoin System Design

Module 15: Technology Stack for Stablecoin Platform

- Selecting front-end & back-end technology
- Smart contract layer: Solidity, Rust (Solana)
- Oracle integration: Chainlink price feeds
- Custody & multisig solutions
- KYC/AML integration for stablecoin issuance
- API gateway & partner integration

Lab Exercises:

29. Design technology architecture for a fiat-backed stablecoin on Ethereum
30. Integrate a Chainlink price oracle into a sample stablecoin contract

Module 16: Designing Stablecoin Issuance System

- Designing stablecoin platform end-to-end
- Mint, transfer, burn lifecycle
- Reserve management & reporting
- Compliance controls: transaction limits, sanctions screening
- Cross-chain bridge design
- Disaster recovery & business continuity

Lab Exercises:

31. Draw stablecoin issuance system architecture with all components

32. Design compliance control flow: KYC check → mint approval → issuance

Day 9: Developing & Simulating Stablecoin Solutions

Module 17: Developing Stablecoin Platform

- Smart contract implementation: mint, burn, pause, blacklist
- Admin dashboard: reserve management UI
- User wallet & onboarding flow
- Back-end: collateral tracking service
- Testing: unit, integration, and invariant tests
- Security audit preparation

Lab Exercises:

33. Deploy a fiat-backed stablecoin smart contract on testnet with mint/burn
34. Build a reserve tracking dashboard connected to the deployed contract

Module 18: Simulating & Evaluating Stablecoin Solution

- Simulating stablecoin platform end-to-end
- Stress testing: de-peg scenarios & liquidation
- Regulatory reporting simulation
- Cross-border payment simulation using stablecoin
- Capstone: present final prototype to peers
- Programme wrap-up & next steps

Lab Exercises:

35. Run end-to-end stablecoin simulation: collateral deposit → mint → transfer → redeem
36. Capstone presentation: demonstrate CBDC, Crypto, or Stablecoin prototype